REPORT DOCUMENTATION PAGE

1. REPORT DATE (DD-MM-YYYY) 2002	2. REPORT TYPE Conference Paper	3. DATES COVERED (From - To) 2000 to 2002

4. TITLE AND SUBTITLE
A Backroom Mission Operations Center for TechSat 21

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62601F

6. AUTHOR(S)
Paul Zetocha

5d. PROJECT NUMBER
8809

5e. TASK NUMBER
LJ

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/VS
3550 Aberdeen Ave. SE
Kirtland ARB, NM 87117-5776

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; Distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The TechSat 21 satellite program is an Air Force Research Laboratory (AFRL) technology initiative which has an objective to demonstrate and validate microsatellite cluster system concepts and enabling technologies. The primary experimental objectives are to demonstrate formation flying algorithms and technologies for clustered satellites, and to demonstrate autonomous cluster and spacecraft operations. TechSat 21 consists of three satellites which will fly in various configurations with variable separation distances. Command and control of a cluster of satellites with multiple heterogeneous experimental objectives poses several challenges from a ground perspective. To assist in operating TechSat 21, AFRL is developing a backroom Mission Operations Center (MOC) which will be capable of performing, among other tasks: planning and scheduling; command generation; state-of-health (SOH) monitoring; telemetry playbacks; fault detection, isolation, and resolution (FDIR); data storage; and payload data analysis. The objective of this paper is to describe the MOC architecture, highlight the key components, and outline its planned operational use.

15. SUBJECT TERMS
TechSat 21, microsatellite, cluster system, formation flying, algorithms

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Paul Zetocha
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified	Unlimited	6	19b. TELEPHONE NUMBER (include area code) (505) 853-4114

i

A Backroom Mission Operations Center for TechSat 21

Paul Zetocha
Space Vehicles Directorate
Air Force Research Laboratory
Paul.Zetocha@Kirtland.af.mil

Abstract — The TechSat 21 satellite program is an Air Force Research Laboratory (AFRL) technology initiative which has an objective to demonstrate and validate microsatellite cluster system concepts and enabling technologies. The primary experimental objectives are to demonstrate formation flying algorithms and technologies for clustered satellites, and to demonstrate autonomous cluster and spacecraft operations. TechSat 21 consists of three satellites which will fly in various configurations with variable separation distances. Command and control of a cluster of satellites with multiple heterogeneous experimental objectives poses several challenges from a ground perspective. To assist in operating TechSat 21, AFRL is developing a backroom Mission Operations Center (MOC) which will be capable of performing, among other tasks: planning and scheduling; command generation; state-of-health (SOH) monitoring; telemetry playbacks; fault detection, isolation, and resolution (FDIR); data storage; and payload data analysis. The objective of this paper is to describe the MOC architecture, highlight the key components, and outline its planned operational use.

TABLE OF CONTENTS

1. INTRODUCTION

For many satellite missions large monolithic satellites are required to satisfy objectives. One prevailing thought is that in many cases clusters of microsatellites can perform similar mission objectives at a lower cost and have performance characteristics that are more optimal while at the same time being less susceptible to failures. Recently various organizations have begun to explore how distributed clusters of cooperating satellites can replace

their larger monolithic counterparts resulting in an overall cost reduction, enhanced mission performance, and increased system fault tolerance. Large clusters of satellites flying in formation are required to have some level of on-board autonomy in order to: fly within specified tolerance levels; perform collision avoidance; perform FDIR; and plan and schedule activities. In addition, from an operations standpoint commanding and controlling a large cluster of satellites can be very burdensome for ground operators. We are addressing these issues by incorporating an on-board cluster manager which will in essence provide the capability to treat the cluster of satellites as a single virtual satellite. From a ground perspective the ground control station must also be able to treat the cluster as a virtual satellite and interact with the on-board cluster manager.

2. MOC ARCHITECTURE

A ground objective is to minimize the manpower required to operate a cluster of satellites and thus the systems within the MOC are designed in order to meet that requirement. For planning and scheduling we are using the NASA/JPL developed ASPEN [1] system which allows translation of high level goals into a sequence of activities to achieve those goals. The activity sequence is then passed to a system hosting the Spacecraft Command Language (SCL) which in turn generates a command transfer frame which is then sent to the cluster of satellites. Prior to sending commands to the satellite they are validated on a testbed which houses PowerPC 750 Engineering Development Unit's (EDU's). SCL is also used to perform limit checking on telemetry mnemonics as well as for detection of anticipated anomalies. A Java frontend to SCL allows access to Telemetry, Tracking, and Control (TT&C) pass replays. For orbit determination and prediction, Satellite ToolKit (STK) and the NASA/GSFC developed GEODE [2] system are being used. STK is also being used to assist in planning both payload data collects and TT&C telemetry downlinks. A Data Center hosting Microsoft Sequel Server is used to store both telemetry and payload data. One of the objectives of TechSat 21 is to be able to treat the cluster of satellites as a single virtual satellite from a command and control perspective. The above tools are thus being developed to satisfy that requirement. The MOC also serves as the interface with the actual spacecraft TT&C and payload ground stations. The MOC architecture and how it interacts with the rest of the TechSat 21 ground system is shown in figure 1.
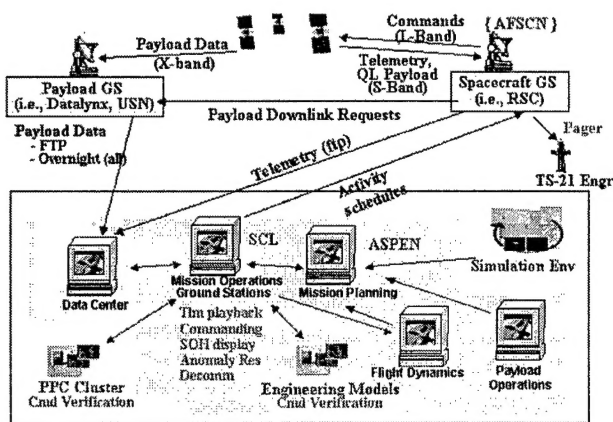
Figure 1: MOC Architecture

The MOC components are shown within the blue shaded region in figure 1. The MOC interfaces with the Research, Development, Test & Evaluation Support Complex (RSC) on Kirtland AFB. The RSC is part of the Air Force Satellite Control Network (AFSCN) and has a primary objective of operating the Air Force's experimental satellites. Command requests are sent from the MOC to the RSC via email and the primary mechanism for retrieving telemetry data from the RSC is FTP. All command and telemetry data is coded in CCSDS packets and sent via S-Band. Due to the large volume of payload data which must be sent to the ground we will be using a commercial ground station provider such as Datalynx or the Universal Space Network (USN). This data will come down via X-band at a data rate of 150 Mbps. Once downlinked to the ground this data will be sent back to AFRL on a tape via overnight delivery.

3. TT&C WORKSTATION

Commanding and controlling a cluster of satellites from the ground poses many challenges including: monitoring state-of-health of the cluster; commanding each vehicle during potentially limited access opportunities; and maintaining relative positioning of the satellites within the cluster. In order to optimize ground operation cost and manpower, different methods are needed to command, control, and monitor the cluster. These methods might include putting more intelligence on-board which would facilitate treating the cluster of satellites as a single virtual satellite and enhanced visualization techniques in order to monitor state-of-health of the cluster.

For a large cluster it is not efficient or cost effective to command satellites on an individual basis. A more efficient method is to send commands to the on-board cluster manager and then have the cluster manager either parse the command string and forward the command(s) to the appropriate satellite(s) or to make some intelligent decision as to the most appropriate action. Two types of ground commanding and controlling are possible.

The first type involves sending up a sequence of commands which are intended for specific satellites. This command block would be sent to the on-board cluster manager where the cluster manager would then parse the command string and send commands to the appropriate satellites.

A second type of commanding involves commands which are sent to the cluster without specific indication as to which satellites in the cluster will ultimately be effected. A hypothetical example might be to issue a command to "observe region x at time y". The cluster manager, based on the status of the satellites at time y, will then determine the appropriate course of action to be taken. This type of commanding requires more on-board intelligence than in the first scenario and has a higher level of risk. Because of the higher level of risk, safeguards need to be put in place to ensure no adverse conditions arise.

To command the cluster of satellites a common frequency will be used with a spacecraft ID used to denote what commands are destined for which satellites. The satellites are flying in close enough formation so that they are all within the same beamwidth. All satellites receive the command but not all will process that command. The operation is somewhat analogous to a TCP/IP broadcast system.

As the core of our Telemetry, Tracking, and Commanding (TT&C) workstation we are baselining the *Spacecraft Command Language* (SCL) from Interface and Control Systems. SCL is a Commercial-off-the-Shelf (COTS) software package which contains an expert system and a command scripting language. It was designed to operate both on-board a satellite and on the ground. This makes it an ideal environment for developing a prototype which contains the cluster commanding and monitoring capability described earlier. This same workstation is planned for use across the program during Integration and Test (I&T).

The TT&C workstation and its interactions with the other MOC workstations is shown in figure 2.
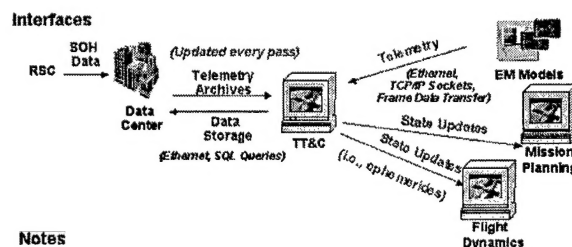

Figure 2: TT&C workstation Interfaces

Raw telemetry data from a satellite pass is run through the TT&C workstation to create Engineering data and to identify any anomalous conditions. This data is then supplied to the mission planning and flight dynamics workstations to perform state updates. The TT&C

workstation is also used to perform command validation by sending commands to the EDU's and validating the correct behavior exists by monitoring the return telemetry.

Expert system rules can be developed and migrated from ground to space as appropriate. Using the rule-based expert system a fault tree for known anomalous conditions can be developed. An initial ground prototype display is shown in figure 3. Although this is for a simplified example it provides cluster level state-of-health monitoring and control.
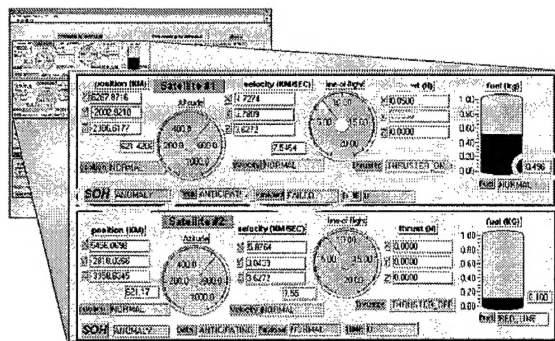

Figure 3: Ground station display

4. MISSION PLANNING WORKSTATION

At the core of mission planning is ASPEN (Automated Scheduling and Planning Environment) which was developed by NASA/JPL. ASPEN works by incorporating spacecraft models, constraints, and objectives into the planning system. ASPEN then generates a sequence of activities which will achieve the desired objectives. Inputs into the planning system would include data such as the mission profile, projected AFSCN and X-band contacts, projected target locations, and current and projected spacecraft State-of-Health (SOH). ASPEN will be used at several levels. At a coarse top level it will be used to generate a one year mission profile. At the next level it will be used to develop two week plans which will be given to the RSC. At the finest level it will be used to generate plans over a 24 hour period. This data will then be used by the RSC to schedule the desired spacecraft activities with the AFSCN. Figure 4 provides an example of an ASPEN display. This figure essentially highlights resources along the vertical axis vs. use of those resources along the horizontal axis.
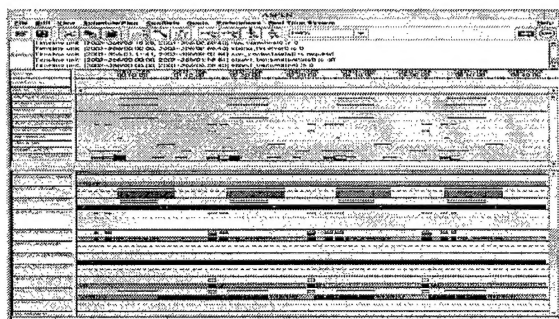

Figure 4: ASPEN

5. FLIGHT DYNAMICS WORKSTATION

The flight dynamics workstation is used to perform several functions. Mission planning uses Satellite Toolkit to plan data collects, access times, and data downloads for both S-band telemetry and the X-band payload data.

A second use of the flight dynamics workstation is to verify that the on-board formation flying algorithms are operating within specifications. For orbit determination we are using the NASA/GSFC developed GEODE system. This is used both on the ground and on-board the satellites. Of specific concern is the relative positioning between satellites. Downlinked ephemeris telemetry data is compared with theoretical data on the ground to ensure the cluster is operating within tolerances.

6. FLIGHT TESTBED

The flight system section of the testbed consists of eight Force PowerCore 6750 boards. The boards have a single PowerPC 750 processor and are housed in a VME chassis. They are connected using 100 Mbps ethernet. In addition each board has two RS-232 interfaces. Each board is running Enea's OSE RTOS. OSE is a message passing operating system well suited for distributed applications.

As can be seen in Figure 5, the flight system interfaces with a simulation environment and with the ground segment of the testbed. The simulation includes spacecraft dynamics, environmental factors, and actuator and sensor models. It provides inputs to the software on the flight boards and receives software outputs to the spacecraft actuators. At present, the simulation is connected to each board via one of its serial interfaces. In the future, the boards will interface with the simulation environment through ethernet. Communication between the ground and flight systems is accomplished through the ethernet and is handled by SCL, which is present at both ends of the interface.
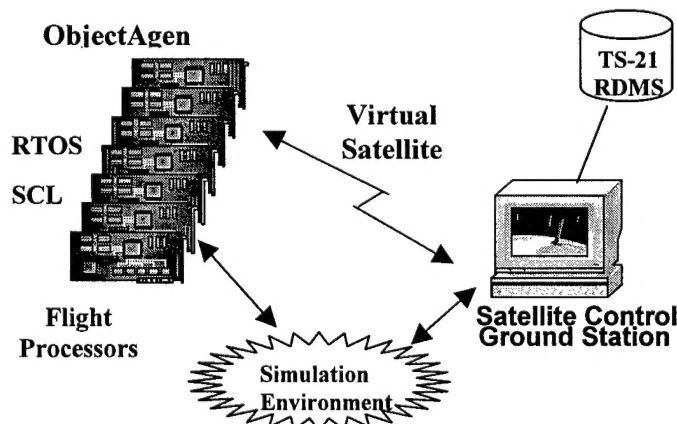
Figure 5: TechSat-21 testbed

The figure shows the three major components of the testbed. The ground system is shown on the right side, the flight system is shown on the left side and the simulation environment is in the middle. The Real-Time Operating System (RTOS) used is OSE and the Remote Data Base Management System (RDMS) is NT SQL Server. The simulation environment will not be discussed because different simulations can be used depending on the required level of fidelity.

The testbed can be used to simulate a cluster of up to eight satellites with the flight software for a single satellite running on each of the processing boards. The prototype system being tested at the present time, however, consists of a three satellite cluster organized in a leader/follower fashion. The leader satellite is known as the cluster manager, it carries software to allow it to make cluster level decisions and to issue commands to the follower satellites. Because the two follower satellites in this system are designated as primary and secondary back-ups to the cluster manager, they carry the same software on-board; however, software pertaining to cluster manager functionality is turned off until the satellite needs to assume the function of cluster manager.

7. COMMAND GENERATION AND VALIDATION

In addition to the many uses already described one of the most important functions of the MOC is for command validation prior to upload to the space vehicles. The primary components used to perform command validation are the TT&C workstation and the engineering development units. The setup used to perform command validation is shown in figure 6.

As shown in the above figure the mission planner generates a sequence of activities. This sequence is then passed to the TT&C workstation which in turn generates a command

Figure 6: command Generation and validation

transfer frame. This command string is then sent to the engineering development units where the commands get invoked in a simulated environment. Satellite telemetry is then returned to the TT&C workstation where it can be used to validate the command that was sent.
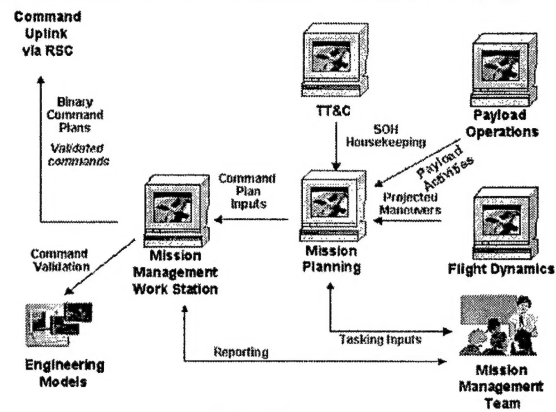
8. CONCLUSION / FUTURE RESEARCH

The development of the MOC is largely still in the preliminary stages. Prototype TT&C and mission planning workstations have been developed but need to be extended. The current TT&C prototype utilizes a limited set of commands and telemetry points, however extensibility to a more complete set of commands and telemetry parameters has factored into the design. In the case of the TT&C workstation it must be extended to contain a complete command and telemetry database. In the case of ASPEN models for all the subsystems need to be added along with all mission operating constraints. The engineering development units are still in the development stage as well are many of the subsystem models and environmental simulations.

5. REFERENCES

[1] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran , "ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling," *SpaceOps 2000*, Toulouse, France, June 2000.

[2] Bauer, F., E.G. Lightsey, S. Leake, J. McCullogh, J. O'Donnell, Jr., K. Hartman and R. Hart. (1995). "The GPS Attitude Determination Flyer (GADFLY): a space-qualified GPS attitiude receiver on the SSTI-Lewis spacecraft." *Proceedings of the ION GPS-95*, Institute of Navigation, Palm Springs, Calif., U.S.A., 12-15, Sept., pp. 555-562.

[3] "TechSat 21: Advanced Research and Technology Enabling Distributed Satellite Systems", *Overview Briefing of TechSat 21*, http://www.vs.afrl.af.mil/vsd/techsat21.

[4] Wyatt J., Sherwood R., Sue M., Szijjarto J., "Flight Validation of On-Demand Operations: The Deep Space One Beacon Monitor Operations Experiment", *Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, ESTEC, The Netherlands, June 1999.

[5] Zetocha P., Self L., "An Overview of Agent Technology for Satellite Autonomy", *1999 FLAIRS Conference Proceedings*, Orlando FL, May 1999.

[6] Paluszek M., Thomas S., Sullivan W., Surka D., "ObjectAgent System Architecture for Autonomous Spacecraft", *Phase I SBIR Final Report*, May 1999.

[7] Zetocha, P., "Satellite Cluster Command and Control", *Proceedings of the IEEE Aerospace 2000 Conference,* Big Sky MT, Mar 2000.

[8] Schetter, T. P., M. E. Campbell, and D. M. Surka, "Comparison of Multiple Agent-based Organizations for Satellite Constellations," *2000 FLAIRS AI Conference*, Orlando, Florida, May 2000.

Paul Zetocha is a Computer Engineer and is currently the lead for the Intelligent Satellite Systems Group of the Air Force Research Laboratory's Space Vehicles Directorate. For the past eight years he has been involved, both as Program Manager and through in-house development, with over a dozen programs related to spacecraft autonomy. He is also the Chairman of the AIAA Intelligent Systems Technical Committee. Mr. Zetocha has received M.S. degrees in both Electrical Engineering and Computer Science from the University of New Mexico with an emphasis in the areas of signal processing and artificial intelligence respectively.